UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release, distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NORDA Report 146	5. MONITORING ORGANIZATION REPORT NUMBER(S) NORDA Report 146
6. NAME OF PERFORMING ORGANIZATION Naval Ocean Research and Development Activity	7a. NAME OF MONITORING ORGANIZATION Naval Ocean Research and Development Activity
6c. ADDRESS *(City, State, and ZIP Code)* Ocean Science Directorate NSTL, Mississippi 39529–5004	7b. ADDRESS *(City, State, and ZIP Code)* Ocean Science Directorate NSTL, Mississippi 39529–5004
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Chief of Naval Operations 8b. OFFICE SYMBOL *(If applicable)* OP-006	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)* Washington, DC 20390–1800	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: 980101 PROJECT NO.: TASK NO.: WORK UNIT NO.:

11. TITLE *(Include Security Classification)*
Recommendations on DMA's Standard Linear Format

12. PERSONAL AUTHOR(S)
Gail Langran, Maura Connor, and R. Kent Clark

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Yr., Mo., Day)*	15. PAGE COUNT
Final	From ____ To ____	July 1986	15

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS *(Continue on reverese if necessary and identify by block number)*
standard data formats, MC&G

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This report stems from a NORDA evaluation of DMA's prototype World Vector Shoreline (WVS). NORDA personnel evaluated the ease of reading the WVS documentation, understanding the file structure, inputting the data from a tape file into program data structures, and plotting the shoreline vectors. Those who participated in the evaluation were previously inexperienced in using Standard Linear Format (SLF), the file structure of the WVS prototype.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ■ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Gail Langran	22b. TELEPHONE NUMBER *(Include Area Code)* (601) 688-4449 22c. OFFICE SYMBOL Code 351

DD FORM 1473, 83 APR

EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

Naval Ocean Research and Development Activity
July 1986 NORDA- **Report 146**

Recommendations on DMA's Standard Linear Format

Sponsored by Chief of Naval Operations (OP-006)

Gail Langran
Maura Connor
R. Kent Clark
Mapping, Charting, and Geodesy Division
Ocean Science Directorate

Foreword

NORDA's Mapping, Charting, and Geodesy (MC&G) Division supports the Oceanographer of the Navy (CNO OP-006) in seeking optimal MC&G digital data formats for Navy users. Among this mission's goals are promoting data processing efficiency, ensuring an uninterrupted and well-maintained data supply, and standardizing MC&G data bases and algorithms. This study evaluates a data format that the Defense Mapping Agency proposes to use for a world vector shoreline.

R. P. Onorati, Captain, USN
Commanding Officer, NORDA

Executive summary

This report stems from a NORDA evaluation of DMA's prototype World Vector Shoreline (WVS). NORDA personnel evaluated the ease of reading the WVS documentation, understanding the file structure, inputting the data from a tape file into program data structures, and plotting the shoreline vectors. Those who participated in the evaluation were previously inexperienced in using Standard Linear Format (SLF), the file structure of the WVS prototype.

Although our study's purpose was to evaluate the WVS prototype, we noted several improvements that could be made to SLF. First, if feature records were placed before segment records, feature classes could be extracted from SLF files in one pass instead of the two passes currently required. Second, several additional header fields would further promote one-pass SLF file processing, including a parameter that states the maximum number of segment vertices in the file and a tally of each feature type used in the file. Third, geographic coordinates are easier to work with analytically if they are ordered as longitude/latitude pairs (rather than the more common latitude/longitude ordering) and marked positive or negative (rather than N, S, E, or W); these changes align geographic with Cartesian coordinates. Federal information processing standards will soon be changed to dictate this coordinate ordering (FICCDC, 1985), which makes it doubly desirable for SLF. And finally, all floating point numbers should be converted to integers, which are faster to exchange among different processors.

We also identified several items of documentation that would be helpful to SLF users. Provision of a file to read the data would have cut several days from our programming effort and made much of the SLF documentation unnecessary. We would also have welcomed a software subroutine to decode the inscrutable 1980-byte Data Set Identifier record. We wrote a rudimentary decoding subroutine and included its output in this report, but a more complete subroutine that fully decodes all acronyms would be better and would minimize the need to refer to the documentation's appendices. Finally, it would be very helpful if DMA could eventually provide a digital FACS look-up table for on-line use.

The most limiting aspect of SLF is its bulkiness, which will become a serious problem as data communications assume increasing importance. To illustrate the economies that are possible when a data format is tailored to its data, we designed an alternate format for the prototype WVS data based on the Federal Interagency Coordinating Committee on Digital Cartography's "Federal Geographic Exchange Format." When we converted the WVS prototype to the alternate format its size was reduced by nearly 50%.

Acknowledgments

This work was sponsored by the Oceanographer of the Navy (CNO OP-006) under Program Element 980101. We would like to thank CDR Richard Blumberg, the OP-006 project manager, and Ms. Mary Clawson of NORDA's MC&G Division for their comments and criticisms.

Contents

1. Introduction 1
- Summary of the WVS prototyping effort 1
- Assumptions 1
- Evaluation method 2
- Order of report 2

2. The WVS's SLF implementation 3
- WVS description 3
- The SLF file structure 3
- Possible compaction methods 3

3. Documentation 6
- Introduction 6
- WVS input software 6
- Data set identification 6
- Clarity 6

4. Content and organization 7
- The header record 7
- Segments 8

5. Summary 8
- Promoting one-pass file input 8
- The SLF documentation 8
- The DSI record 9
- Coordinate coding 9
- Use of integers 9

6. Bibliography 9

Appendix: An alternate space-saving format 11

Recommendations on DMA's standard linear format

1. Introduction

SUMMARY OF THE WVS PROTOTYPING EFFORT

In FY 85, a NORDA study of naval digital mapping, charting, and geodesy (MC&G) data requirements revealed that a large number of Navy systems and commands currently use or plan to use a world vector shoreline. The CIA's World Data Banks I and II (WDB I and WDB II) are supporting most current users. Because continued use of WDB I and WDB II by the Navy was deemed inadvisable for various reasons, the Oceanographer of the Navy (CNO OP-006) requested that the Defense Mapping Agency (DMA) produce a World Vector Shoreline (WVS). Table 1–1 shows a preliminary list of systems that will use the WVS. Table 1–2 lists the WVS's required characteristics (CNO OP-006 memo, ser 006C/50322906, 5 August 1985).

In an effort to save production costs, DMA developed a method of deriving vector shoreline data from its Digital Land Mass Blanking data, a land mask gridded at 3 arc second intervals. A small area covering portions of the Delaware, New Jersey, and New York shorelines was processed by contouring the land/sea interface and generalizing the vertices to several different resolutions. State boundaries and names were added to prototype the international boundaries and country names that will be included in the final data set.

In March 1986, DMA released the prototype WVS for four groups to evaluate: NORDA, Naval Undersea Systems Command, Naval Air Development Center, and the Joint Cruise Missile Program Office. As the Navy's MC&G research laboratory, we at NORDA decided to use the three weeks available for this project to assess the data set's processing efficiency and the suitability of its content. No time was allotted to assessing data quality, although such a study is highly recommended.

The WVS data is structured in DMA's Standard Linear Format (SLF) and the three feature types contained in the data set (shorelines, boundaries, and text placement) are assigned standard Feature Analysis Coding Standard (FACS) codes. The usability of SLF outside of DMA has been hotly debated; this evaluation, however, was one of the first tests of SLF's practicality as an exchange format.

Table 1–1. Potential Navy users of WVS. Computer models and word sizes are listed when known.

System	Computer Model	Bits/Word
APP		
ACDS	UYK-43	32
ASWOC (baseline)	CP901	8 (64K machine)
ASWOC (upgrade)	undetermined	undetermined
CV-ASWM	UYK-7 (later, UYK-43)	32
CCS-MK1	UYK-7	32
DSAT	VAX 11/780	16
DUET	VAX 11/780	16
FHLT	UYK-7	32
HYCAT	UYK-44	16
ICAPS	(15 different models)	
NEAT	VAX 11/780	16
NISC-OFM	VAX 11/750	16
P3 UPDATE	CP901 (later, AYK-14)	
POST		
SACC	Victor AN-ASQ114V	8 (128K machine)
SEABASS	VAX 11/780	16
SEANYMPH	UYK-20	16
SEA WATCH II	CYBER 170/730	60
SEA WATCH III	undetermined	
SOCC		
STT		
TERPES	CP-808 (later UYK 43)	30 (later, 32)
TESS	HP 9020A	32
TWCS	UYK-64, UYK-19	8 (later, 16)

Thus, although our main study goal was to determine an optimum WVS format for naval data users, we also examined the SLF standard with interest. Our suggestions have been compiled in this report.

ASSUMPTIONS

Our evaluation was based on a set of assumptions that impact data set design and packaging.

- Network communications will continue to grow in importance.
- The WVS will supplant WDB II in the Navy, which means that systems now using WDB II will need to convert their software to accept WVS.

Table 1–2. Required WVS characteristics.

• The WVS must use a minimum number of points to display the shoreline at the desired scale, since some systems have limited storage and processing capacities.
• The WVS must support output at scales ranging from 1:250,000 to 1:12,000,000.
• The WVS must use a vector format.
• The WVS must have an accuracy comparable to paper products.
• The WVS must identify the shoreline's land and water sides to allow color fills.
• The WVS must include international boundaries that are maintained in a current condition.
• Disputed boundaries must be identified.
• Country labels must be associated to international boundaries.
• The WVS must be compatible with DMA's DTED, DFAD, HOD, PPDB, and the future digital bathymetric and contour data bases.
• The WVS may be blocked by geographic areas if suitable overlap exists between areas to permit operations at block edges.
• The WVS must embed certain data characteristics (particularly feature tallies and other data size estimates) to promote automated data entry.
• A programmer's appendix to the documentation must supply illustrations and examples of data content.
• Software programs must be provided with the WVS to assist users in reading the data.

- The WVS will support a broad spectrum of naval applications, but not navigation.
- The WVS exchange format does not need to be the same as the master WVS file format.
- A new WVS user will be born every minute.

The final assumption is far from facetious; the quality (or crudeness) of WVS documentation and packaging will have a major impact on the Navy's data processing costs because of the potentially large number of WVS users. The second assumption underlines this need for carefully developed documentation: central, one-time development of subroutines or specific instructions to help systems convert from WDB II to WVS is highly recommended. The alternative—redundant development of conversion methods by different users—would be far more expensive.

The second-to-last assumption is likely to be controversial, but data processing efficiency demands it. Although we began the study assuming that the master and exported WVS files would be identical, as we probed the SLF structure we found much to recommend its internal use by DMA but little to recommend it as a WVS exchange format. SLF was designed to allow a broad spectrum of complex geographic information to be encoded. The all-purpose format entails a large number of blank spaces (over 51% of the WVS prototype's first file is spaces). More important, SLF's complex capabilities are not used by the prototype WVS data, which is essentially "spaghetti" data (e.g., data with no description of geographical or topological relationships).

The rest of the assumptions affect data set design and influenced our evaluation. The immediate use of network communications to transfer WVS data is unlikely; however, WVS's size should be minimized in anticipation of such an event. Display for non-navigational use allows a certain latitude in data accuracy. Data quality problems have already been noted by DMA (some shorelines cross themselves), but many will not be noticeable at the expected display scales. Finally, the assumed broad spectrum of WVS applications encouraged us to develop an evaluation method that would address a wide range of geographic data processing problems.

EVALUATION METHOD

By envisioning prospective WVS applications (i.e., tactical planning, enhancement of environmental data displays) we developed a short list of important data processing tasks. Carrying out the tasks on our list made us quite familiar with the prototype and allowed us to study its usability in several different lights. Table 1–3 is a list of the manipulations that were performed at NORDA.

Table 1–3. Summary of NORDA's WVS evaluation. Asterisks follow tasks that were in process at press time.

Read documentation, become familiar with SLF, read tape
Plot the coastline in hardcopy
Clip and plot a geographic window from the file
Perform color "fills" on the land and sea pixels*
Extract one feature type and write to a subfile

ORDER OF REPORT

The rest of this report discusses some of the discoveries made during the study concerning SLF.

- Section 2 describes the WVS implementation of SLF and discusses problems with the bulkiness of SLF data sets. Several ways of compacting the WVS data within the confines of SLF are suggested, and a compaction method that departs from SLF is described.
- Section 3 suggests helpful ways to document SLF files.

- Section 4 discusses changes to SLF file content or organization that could make files easier or more efficient to process.
- Section 5 compiles our suggestions concerning SLF.
- Section 6 cites references.
- The Appendix describes a space-saving format for the prototype WVS data designed at NORDA.

2. The WVS's SLF implementation

WVS DESCRIPTION

The WVS prototype was delivered at three different resolutions on one tape volume. Each data resolution comprises one file set. All three file sets are coded in SLF. Table 2-1 describes the dimensions of File Set One.

Table 2–1. WVS prototype file set description. The WVS prototype contains three file sets, each containing three files, to support plots of various point densities. This table describes File Set One, which is used throughout the report when an example is needed for discussion.

File set name	SLFFUL.GO
No. vertices (approx)	42000
Pts/mile (approx)	14
NUMBER OF BYTES:	
Volume header	960
DSI Record	1980
Segment Records	734580
Feature Records	219780
Text Records	1980
Volume Trailer	480
Total Bytes	959760
NUMBER OF SEGMENTS AND FEATURES	
Segments	1215
Segment Records	371
Features	1215
Feature Records	111

THE SLF FILE STRUCTURE

Figure 2-1 is excerpted from DMA's draft SLF documentation (second edition, 18 March 1985). In brief, SLF files are divided into 1980-byte physical blocks, which are also logical records. Each logical record is one of four types: Data Set Identifier (DSI), Feature (FEA), Segment (SEG), or Text (TXT). A block's first eight bytes indicate which record type the block contains. Every SLF file has at least one of each record type (even if a record type is not needed); a block that is only partially filled is padded with blanks to 1980 bytes.

The first SLF block is always a DSI record, which contains all data set parameters, including its origin, security classification, coordinate descriptions, history, and accuracy. SEG records, which contain all feature vertices, follow the DSI record. After the SEG records come the FEA records, which describe each feature's attributes and contain keys to feature vertices stored in the SEG records. TXT records are last; they provide space for optional free-format textual information.

The purpose of separating vertices from features is to avoid storing line segments redundantly. An example is when a river, a political boundary, and two soil type polygons share a line segment. Early geographic data processing technology would cause such a line to be stored four times: once for the river, once for the political boundary, and once for each of the two soil polygons. Such files were rife with duplicate line segments, which often were slightly mismatched due to repeated digitization. An SLF file stores the line segment once. The river, boundary, and soil polygon records contain keys to all their component segment records. Thus, SLF efficiently stores complex geographic data.

POSSIBLE COMPACTION METHODS

SLF provides many elegant data storage options, but it is bulky. The prototype WVS is inordinately long for a file comprised of simple shoreline and boundary vectors. The length comes from spaces used to pad fields and records, and from feature/segment pointers. The following subsections describe the WVS implementation of SLF and discuss compaction options.

Segment records

Although the WVS includes no Z values, a 6-byte Z-value field is included for each vertex, which increases the total space used by segment records by nearly one-third. Assuming an average of 50 vertices per segment, the total space savings accrued by omitting the Z-value space for all 1215 of File Set One's segments is 365 kbs, or 24.5% of File Set One's total space.

SLF allows Z values to be omitted. The DSI record's "vertical units of measure" parameter, if left blank, indicates that SEG records contain coordinate pairs (x, y) rather than triplets (x, y, z). The WVS prototype fills the "vertical units of measure" field with "M" for meters, erroneously implying that a Z value is needed.

Feature records

Feature records are divided into subrecords that describe individual features—shoreline segments, boundaries, or

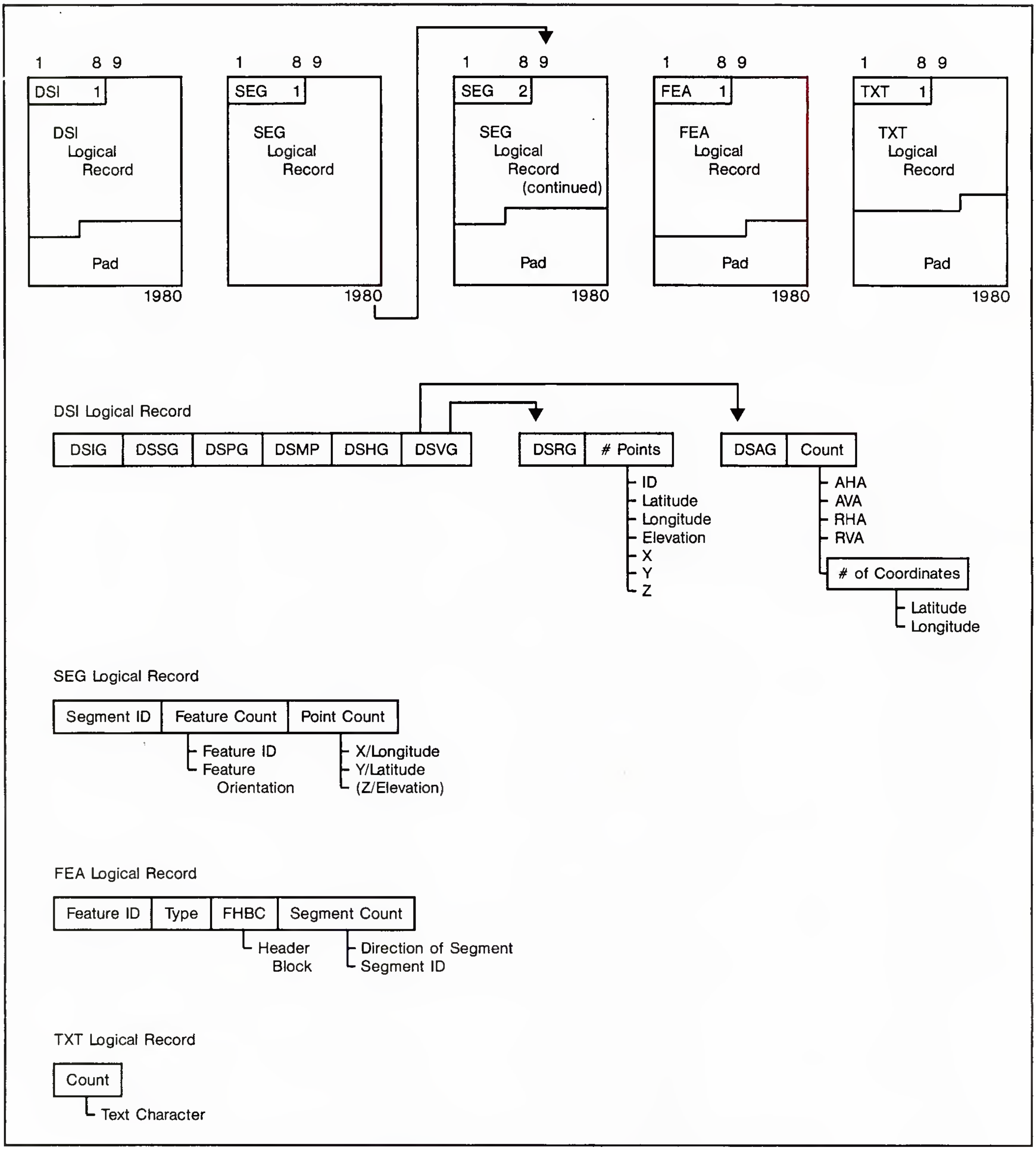

Figure 2–1. SLF physical and logical record structure (DMA, 1985).

placenames. A feature subrecord is comprised of 179 bytes: 10 bytes for parameters, a 160-byte feature header, and 9 bytes to reference the feature vertices stored elsewhere in segment records. Thus, 89% (200 kbs) of the 220 kbs comprising File Set One's feature records are devoted to highly redundant feature headers (Table 2–2).

Three ways exist to compress feature subrecords. The amount of compression achieved is inversely related to

Table 2–2. Information stored in each subrecord's 160-byte feature header. Most headers contain 96 blank spaces (except "position of text" subrecords, which uses some of this space for the text label). For the most part, the information varies between but not within feature types (e.g., one attribute is true for all shorelines). No documentation was provided for header fields, hence, the table's "unknown" values.

Size (bytes)	Value	Use	Redundant?
10	500000	data scale?	yes
8	01	unknown	yes
10	0	unknown	yes
4	0	unknown	yes
4	U	classification	yes
9	(variable)	feature code	varies with feature type
13	(variable)	unknown	varies with feature type
66	(empty)	N/A	yes
36	(0.0000/text)	(unknown/ country name)	varies with feature type
TOTAL: 160 bytes			

the resulting file's adherence to SLF. The first adheres to the letter, the second to the spirit of SLF. The third, which departs entirely from SLF, is discussed in the next subsection.

Remaining strictly within SLF, 146 kbs could be saved in File Set One alone by reducing the 160-byte feature header to 40 bytes. Of the 40-byte header, six bytes would be used for the feature code, leaving 34 bytes empty for country names. The handful of approved country names that exceed 34 characters are easily abbreviated; in fact, abbreviation of any name exceeding 15 or 20 characters is preferable for digital display.

A second compression method is to dispense with feature headers altogether (a file comprised of only three simple feature types scarcely seems to warrant them) but leave the rest of the SLF format untouched. Instead of feature headers, a 5-byte feature ID field and approximately 20 bytes for attributes would be added to all subrecords. Attributes are not needed for shorelines, but boundaries need a maintenance date, disputation status, and pointers to their bounded countries. Countries would use the attribute space for the placename (this assumes that shortened country names are used). Thus, the 42 bytes needed to store the necessary information using the first compression method (40 bytes for the feature header and 2 bytes for the feature header block count) is reduced to 27, saving an additional 164 kbs overall. Table 2–3 summarizes the effects of the first and second compression methods.

Table 2–3. Impact of SLF-based compression methods on feature subrecord header data.

WVS Prototype		Method One		Method Two	
Header blocks	2	Header blocks	2	FACS code	6
Header	160	Header	40	Attribute	21
Total Bytes	162	Total Bytes	42	Total Bytes	27

Departure from SLF

The third compression method departs completely from SLF. SLF requires that feature vertices be segregated from other feature data in SEG records. In the WVS prototype, SEG records are divided into subrecords that contain the vertices for one shoreline, boundary, or text feature (whose attributes are stored in a FEA subrecord). SEG and FEA subrecords contain key values to cross-reference features to vertices. Because SLF was designed for data that lacks a one-to-one feature-to-segment correspondence, it provides space to store information that does not apply to a spaghetti WVS (Table 2–4).

Table 2–4. SEG and FEA record cross-referencing. Each feature references one (and only one) segment: feature n keys to segment n. Feature orientation and segment direction are the same throughout the file. Thus, 31 bytes/feature (or 37 kbs in File Set One) are extraneous to the WVS prototype data.

SEG
Key value (6 bytes)
Number of features referencing this segment (2 bytes)
Keys of all features referencing this segment (6 bytes each)
Orientation of each feature referencing this segment (1 byte)
FEA
Key value (6 bytes)
Number of segments comprising this feature (3 bytes)
Keys of all segments comprising this feature (6 bytes each)
Direction of each segment comprising this feature (1 byte)

A great deal of space and processing time could be saved in the prototpe WVS file by not separating features from their segments. Because the WVS has a one-to-one correlation between features and segments (e.g., feature 1215 is referenced to segment 1215) and no WVS feature references more than one segment, placing feature *n* in the same record as segment *n* allows us to dispense with

the data exhibited in Table 2–4. When the WVS is upgraded to encode geographic relationships, separation of features from segments may still not be the most economical way of storing feature vertices.

Results of WVS file compression

As part of our evaluation we designed a file format based on the compaction strategies discussed in this section (Appendix). The alternate format illustrates the efficiencies that are gained by tailoring a format to the data it will hold. Our format borrows the field/subrecord/record terminator strategy used by the Federal Geographic Exchange Format (FICCDC, 1986), which allots file space according to the length of a data element, not according to a fixed specification as does SLF. Even after adding new fields for needed data elements (i.e., boundary attributes and feature tallies), the resulting file was compacted to nearly one-half its former size.

3. Documentation

INTRODUCTION

The documentation provided with the WVS prototype was generic by necessity: prototype documentation has not yet been developed to accompany the prototype WVS. Therefore, our comments in this section are geared toward helping DMA to develop appropriate documentation standards.

WVS INPUT SOFTWARE

A READ program or subroutine should be provided with all files to make access easier. Writing one is not a trivial task (the READ program for the WVS prototype took nearly 20 hours). The READ programs and any other data processing programs provided by DMA could be placed in TXT records that precede the DSI record.

DATA SET IDENTIFICATION

The documentation should help to minimize the amount of time spent determining what is in the data set. DMA's data users should not be forced to inch through an SLF DSI record (Fig. 3–1), counting spaces and flipping to the SLF documentation appendices to decode the header information.

Three methods can be used to reduce the amount of time the user must spend lost in the DSI. First, much of the administrative information could be eliminated for export (DMA would retain this information in its master files), which would make the DSI easier to read without altering its structure.

Second, WVS users could be provided with a DSI-decoding subroutine. Output from a NORDA-generated DSI-decoding subroutine is shown in Table 3–1. The ideal DMA-provided subroutine would decode all DSI abbreviations and list the DSI information in an easy-to-read format.

Third, and best, a less complex header record could be designed for exported data products. Two levels of header information could be identified: information needed by all users at a glance (i.e., title, coordinate conversion information, originating agency) and information needed only for troubleshooting and diagnostics. The first type of information would be annotated and easy to read. The second type of information would be more compressed and the user could simply ignore it unless problems occur.

CLARITY

Employing a professional editor to clarify SLF documentation could cumulatively save DMA's data users a great deal of time and frustration. Some WVS processing problems experienced by NORDA and other WVS evaluators were attributable to problems in interpreting the documentation. In many cases, vital information is deeply embedded in trivialities.

```
DSI     1DSIGWVS   WVS FULL RESOLUTION    1860186 1                                      DSSGU   OADR   UNCLASSIFIED
                                   DSPGGEOSEC   .10WGCWGCM   000.1MSL MHW 37550000N076050000W
      37550000N076050000W41001570N073000000W  1215     3   1212     0   1215                                     DSMP
TC075000000W                              500000                                      DSHG  30DMA/SLF        8503   2 2
USDMAHTCACDDS     ACDDS          0    0    0    0      00000                                         DSVG   646
 1469                                DSRG  16      138000000N076000000W                                    239000000N07
6000000W                           340000000N076000000W                             441000000N076000000W
              538000000N075000000W                                 639000000N075000000W
  740000000N075000000W                             841000000N075000000W                                938000000N07400
0000W                        1039000000N074000000W                       1140000000N074000000W
            1241000000N074000000W                                   1338000000N073000000W                            1
439000000N073000000W                         1540000000N073000000W                              1641000000N07300000
0W                   DSAG00
```

Figure 3–1. WVS prototype DSI record.

Table 3–1. Output of NORDA's DSI-decoding subroutine. The ideal DMA subroutine would contain look-up tables to decode all data type, datum, etc., codes.

```
Header field = "DSI    1"
Product Type = WVS
Data Set ID = WVS FULL RESOLUTION
Edition no. = 1

Dates of Compilation and Maintenance = JAN 86, JAN 86
SLF and DMAFF Version Dates = 0 0 0    0 0 0
                                    (Not given in 1st ed.)
Security classification = U

Data Type = GEO
Horizontal units of measure and implied decimal = SEC, 0.10
Geodetic Datum = WGC
Ellipsoid = WGC
Data generalization code = 0

Latitude, Longitude of Origin = 37.9167 −76.0833
SW corner = 37.9167 −76.0833
NE corner = 41.0044 −73.0000

Number of Features: point, line, area = 3, 1212, 0
Total number of Features and Segments = 1215, 1215

Map Projection Code = TC (Transverse Mercator)
Scale = 1: 500000
Projection Parameters = −75.0000  0.0000  0.0000  0.0000
```

One such problem was our belated discovery that WVS coordinates, listed in integer seconds, require division by one-tenth for normalization. We eventually found a DSI parameter called "Horizontal Resolution Units" with a value of 0.10, which the documentation defines as the "number of units of measure which constitute the least count of the horizontal coordinate system." Similarly clumsy wordings obscure meanings throughout the documentation.

4. Content and organization

THE HEADER RECORD

Reductions

An exported file's header needs only a subset of the information contained in the master file's header. Table 4–1 summarizes data we felt could be omitted from the WVS's DSI record.

Additions

An exchange format should assume sequential processing and promote one-pass input. One-pass tape input means that when the last record is reached, all data has been stored in the desired form with no rewinding or other backspacing. One-pass input using network communications means data is received and stored "on-the-fly." The alternative is to devise a temporary holding file for data, which is structured later according to program requirements.

To achieve one-pass input, careful thought must be given to what information will be needed at which point in the input process. For the WVS this means including certain data set parameters in the header record. First, the header must state the number of vertices in the data set's longest line segment so arrays can be dimensioned correctly in computer memory. If this number is missing (as it is from SLF files), then a programmer must guess a value and risk its being too small (which crashes the program) or too large (which wastes space); or, the programmer must settle for two-pass input by running a vertex-counting program on the data prior to input.

The second required set of parameters is a tally of each feature type contained in the file. In the case of the WVS, which always has the same three feature types, the header could simply provide three numbers (e.g., "1922, 5, 8," meaning 1922 shorelines, 5 boundaries, and 8 country names).

Table 4–1. Summary of unnecessary DSI information.

Data Set Identification Group. Most of this data could be combined into one long fixed-format string. Then, the information is preserved but the user can be instructed to ignore it unless the data set's origins must be traced. The exception is a clear English title, set off by asterisks and spaces, that says something like "DMA World Vector Shoreline, 1:500,000, March 1985, Edition 1."

Data Set Security Group. If the WVS is unclassified this entire group could be omitted.

Data Set Parameter Group. Vertical units of measure, vertical resolution units, vertical reference system, and sounding datum are not needed. The origin's Z value is also not needed. The header should provide separate tallies of shoreline vectors, boundaries, and text positions in addition to the overall feature and segment tallies currently provided.

Data Set Map Projection Group. All this information should be omitted from the WVS to avoid confusion.

Data Set History Group. Not needed.

Data Set Variable Field Address Group. Not needed.

Data Set Registration Points Group. Not needed.

Data Set Accuracy Group. Only a horizontal accuracy statement is needed.

SEGMENTS

Segment order

How to order the WVS segments to promote easy windowing is one of the less tractable WVS data structuring problems. A good way to order WVS segments would be to group them into grid cells, causing nodes to occur at grid intersections and segments to be as continuous as possible within grid cells. UTM zones (6^0 x 8^0) could be a useful cell unit for small-scale (1:3,000,000 and smaller) files. Smaller cells should be used for larger-scale files.

Coordinate coding

The Cartesian coordinate system used for most mathematical analysis has an origin (0,0) from which measurements are made in a positive or negative x or y direction. Cartesian coodinates are ordered (x,y). Conversely, geographic coordinates are ordered (y,x), or (latitude,longitude). Geographics state positive and negative x direction as east and west, respectively, and positive and negative y direction as north and south, respectively. Finally, geographic coordinates are often provided in degrees, minutes, and seconds, rather than in decimal values.

Because most analysts work instinctively with Cartesian, but not geographic, coordinates, every effort should be made to align the geographic to the Cartesian coordinate system for analytical use. Recognizing this problem, FIPS PUB 70, "Representation of Geographic Point Locations for Information Interchange," will soon be changed to dictate the transmission of geographic coordinates in a (longitude, latitude) ordering. We recommend that WVS adhere to the new standard. We further recommend that N, S, E, W always be stated as positive and negative departures from the origin or that coordinates be normalized to binary fractions, as recommended in the proposed Canadian Map and Chart Data Interchange Format (OMNR, 1985).

5. Summary

A few minor changes, summarized below, could help make SLF a more versatile file format.

PROMOTING ONE-PASS FILE INPUT

One-pass input allows a file to be read into computer memory from beginning to end with no need to rewind or otherwise backspace. It is helpful for tape input and vital to efficient network communications. The current SLF format requires at least two, and (depending on the application) possibly more, passes through the data for file input. Some illustrations follow.

Extracting several feature types into a subfile

A user may wish to extract only cultural features from an SLF file comprised of both cultural and natural features. To accomplish the extraction, the user first passes through the SLF file to the FEA records and collects all cultural features, noting their keys. He then rewinds (or otherwise backs up) to the SEG records, where he matches the keys of the collected features to the segment keys.

This two-pass process can be reduced to one pass by simply placing all SEG records after the FEA records. Then, after collecting the qualifying features and their keys, the user proceeds sequentially to the SEG section and collects the matching segments.

Dimensioning program structures for file input

Features. Many applications require a separate array to be dimensioned for each feature type (i.e., one for populated places, one for streams, etc.). To dimension arrays of the proper length, current SLF users must first pass through the file and count the occurrences of each FACS code in the file, then pass through a second time to input the features. We feel the better alternative is to place such a tally in SLF's DSI record. It would be easy to count the features as the file is being created and supply this information to the user.

Segments. Data structures must also be dimensioned for line segments. Two values must be known to efficiently create the array space needed to store the segment vertices: the total number of segments (which SLF currently provides) and the longest segment's length. To get the second value the user must pass through the file once to find the longest segment, then pass through a second time to input segment vertices.

The text record

Free-format text explanations would be far more useful in front of a file than near the end. If placed in front, the text records could hold tape documentation (e.g., DMA'S WORLD VECTOR SHORELINE (1:500,000), Edition 2, December 1986), programs (particularly a READ program and possibly a DSI-decoding program), and messages to users.

THE SLF DOCUMENTATION

Perhaps the best aspect of the SLF documentation we received from DMA was a partial printout of portions of WVS File One to step through. A figure of this type should be made a permanent part of SLF's documentation; it is hard to appreciate the format without an example. The current general graphic examples make a good supplement to an explicit example.

We would prescribe a professional editor to clear up clumsy wordings in the documentation, since some DSI parameters are not defined well enough for DMA outsiders to understand their purpose. Presumably, those who need them understand the explanations, but fretting over them slowed us down. Particularly impenetrable was the wording that defined horizontal and vertical resolution units, the difference between the two sets of accuracy statements (one in the data set's history group, one in the address group), and the purpose of the match/merge parameters. Some of our comprehension problems were exacerbated by the fact that the WVS prototype's header appeared to be in error in several fields (e.g., inclusion of a map projection despite the use of geographic coordinates, and inclusion of geographic but not x and y registration points).

A more comprehensive glossary would also be very helpful, due to the number of acronyms used in the SLF documentation. A minor point is the consistent use of the term "fpi" throughout the SLF documentation in places where one would expect to see "bpi." If it is not a typographical error, then "fpi" should be defined.

THE DSI RECORD

All codes that pertain to a single category of information should be the same length. In SLF, units of measure range from one to three characters. Datums range from three to four characters.

It may be useful to give all four corners of the data set to allow for non-square data windows (i.e., nongeographic coordinates in a conic projection) and data windows with x and y axes that do not coincide with lines of latitude and longitude.

When applicable it may be useful to include a "pole" parameter for conics (e.g., Lambert). Example, +1 if over the north pole; −1 if over the south pole. Standard parameters do not always make this distinction clear.

COORDINATE CODING

As discussed in the previous chapter, geographic coordinates are easier to work with when they are aligned with Cartesian coordinates. This means that the geographics should be expressed as (longitude, latitude) pairs rather than the current (latitude, longitude) convention, that north, south, east, and west be encoded as positive and negative departures from the origin, and that decimal values be used throughout. Other government standards are leaning in this direction.

USE OF INTEGERS

Floating point data can be difficult to convert between different processors due to word size and internal handling differences. With minimal effort SLF files could be purged of all floating point numbers by converting them to integer values and providing the conversion factor.

The WVS prototype has only two floating point numbers in the DSI record (one for horizontal, the other for vertical resolution units). If (as we have assumed) these numbers are the vertex conversion factors, they could also be expressed as the reciprocal value by which to divide the data to convert it back to floating point (e.g., "10" instead of "0.10").

6. Bibliography

Central Intelligence Agency (1977). *WDB II General Users Guide.* PB 271 869, Washington, D.C., July.

Defense Mapping Agency (1985). *Standard Linear Format for Digital Cartographic Feature Data.* Second Edition, Washington, D.C., 18 March.

Defense Mapping Agency (1985). *Feature Analysis Coding Standard.* Appendix VIII, First Edition, Washington, D.C., 30 July.

Federal Interagency Coordinating Committee on Digital Cartography (1986). *Specification for a Federal Geographic Exchange Format.* 31 January.

Magee, Ronald L. and Claudia K. Slaton (1985). *High Resolution Data Base Evaluation Final Report.* Naval Training Systems Center, Orlando, Florida, September.

Mandico, Nancy et al. (1984). *The Prototype KBGIS Final Report.* PAR Technology Corporation, Washington, D.C., November.

Ontario Ministry of Natural Resources (1985). *MACDIF: Map and Chart Data Exchange Format.* Surveys and Mapping Branch, Ottawa, Ontario, Canada, June.

Appendix: An alternate space-saving format

This alternate format was designed to hold the prototype WVS's spaghetti vectors to illustrate that a format tailored to its data is far more space- and processing-efficient than a generic format. The new fields added for boundary attributes and feature and segment tallies are balanced by the elimination of some DSI information. Thus, the nearly 50% space savings of this format over SLF is fairly indicative of the compression level achieved. A description follows.

The format was modeled on the FICCDC's Federal Geographic Exchange Format. Variable-length records with a maximum of 80 characters (for easy display on a video terminal) are used. File, record, subrecord, and character string delimiters are defined to control data input. Format specifiers are included with the file for easier reading, although the file may be read in free format if preferred. All data values are either character strings or integers.

Header information

The first three records (total 240 bytes) substitute for the DMA prototype's 1980-byte DSI record. DSI values that we thought were unnecessary and most filler spaces have been omitted to save space. Table A–1 describes the three records that comprise the data set's logical header.

Delimiters that describe file, record, subrecord, and character string endings are stated in the header's "DLIMIT" field. In this example, records are terminated by "&," subrecords by "/," files by "$," and variable-length character strings by "@." Delimiters aid in searches for specific features, allow programs to skip over unwanted information, and allow records containing less than 80 characters to be shortened to their actual length.

Table A–1. Alternate header format.

Variable Name	Format	Explanation
RECORD ONE: 80 bytes		
DSID	A20	Data Set ID
ENUM	I3	Edition number
CDATE	I4	Compilation date
MDATE	I4	Maintenance date
SDATE	I6	SLF version date
DLIMIT	A4	Delimiter codes: file, record, subrecord, string (e.g., $&/@ = file $, record &, subrecord /, string @).
BLANK	A39	Comments and further description
RECORD TWO: 80 bytes		
DTYPE	A3	Type of coordinates*
DLNGOR	I9	Longitude of origin
DLATOR	I8	Latitude of origin
DLNGSW	I9	Longitude of SW corner
DLATSW	I8	Latitude of SW corner
DLNGNE	I9	Longitude of NE corner
DLATNE	I8	Latitude of NE corner
HORZUM	A3	Horizontal units of measure*
HORZFP	I6	Floating point placement*
ISCALE	I9	Scale reciprocal
BLANK	A21	Filler space for future use
RECORD THREE: 80 bytes		
NCODES	I6	Number of FACS codes in file (e.g., # feature types)
NSEG	I6	Number of segments in file
MSEGVRT	I6	Maximum possible vertices per segment
BLANK	A61	Filler space for future use
EOR	A1	

*NOTE: If DTYPE = GEO, HORZUM = SEC, and HORZFP = 10, the file's coordinates are in (longitude, latitude) pairs measured in decimal seconds. "HORZFP = 10" denotes that coordinates were transformed to integers by multiplying by ten, and can be restored to world coordinates by dividing by ten. All coordinates are referenced to the origin (given in RECORD TWO).

Data records

Country names. The first series of records following the header are country names, stored as FACS code 9A005. Each record is comprised of subrecords that contain a country's character string and centroid in (longitude, latitude) ordering. A unique country key cross-references countries to their boundaries (but not boundaries to their countries, which is beyond the scope of a spaghetti data file). Use of a character delimiter allows a string to occupy from 0 to 39 spaces. Table A–2 describes country name records.

Boundary data. All boundary data is grouped following the country names data. Boundary segments are defined as continuous sections of international boundary lines with identical attributes, i.e., same maintenance date, disputation status, and with the same two adjacent regions.

The boundary data group is headed by the boundary FACS code, the total number of boundary segments, and the maximum number of boundary segment vertices. Although the file header states the maximum number of segment vertices for boundaries and shorelines combined, providing individual maximum values for each feature type allows space-conscious programmers to shorten boundary segment arrays, which can be considerably shorter than shoreline segments.

Table A–2. Country name records. The first three variables head the country name group. The next set of five variables repeats until all country names are encoded. A desirable upgrade would be to provide several versions of each country name: the formal name, one or more commonly used names, and an abbreviated name for display labeling.

Variable Name	Format	Explanation
FACS	A5	Text string feature code (9A005)
NSTRINGS	I6	Number of text strings
SUBREC	A1	Subrecord delimiter (/)
CTYKEY	I4	Unique country key
STRING	A40	Country name followed by string delimiter (@)
LNG	I6	Longitude of country centroid
LAT	I6	Latitude of country centroid
SUBREC	A1	Subrecord delimiter (/)
(repeat the preceding four variables NSTRINGS times)		
EOR	A1	End of record delimiter (&)

Following the boundary data group heading, boundary segments are listed with attributes preceding the vertices. Table A-3 describes the boundary data group.

Shoreline segments. Shoreline records are structured much the same as boundary records except they are not tagged with attributes. The group is headed by the shoreline FACS code, the total number of shoreline segments, and the maximum number of shoreline segment vertices. All shoreline segments follow the heading as simple strings of (longitude, latitude) pairs separated by delimiters. Table A-4 describes shoreline records.

Table A–3. Boundary records. Each boundary segment separates no more than two countries and is tagged with attributes describing its most recent maintenance date and disputation status.

Variable Name	Format	Explanation
(header)		
FACS	A5	Feature code
NSEG	I6	Total number of boundary segments
MAXVERT	I6	Maximum number of boundary segment vertices
SUBREC	A1	Subrecord delimiter (/)
(segments)		
NVERT	I6	Number of vertices in this segment
DSPUTE	I1	Disputation status
MAINT	I6	Maintenance date
CTYLEFT	I4	Key to country on left of boundary segment
CTYRGHT	I4	Key to country on right of boundary segment
LNG	I6	Vertex longitude or X value
LAT	I6	Vertex latitude or Y value
(vertices repeat NVERT times)		
SUBREC	A1	Subrecord delimiter (/)
(segments repeat NSEG times)		
EOR	A1	

Table A–4. Shoreline records. Because shorelines need no attributes in this spaghetti structure, shoreline records are simple lists of vertices.

Variable Name	Format	Explanation
(header)		
FACS	A5	Feature code
NSEG	I6	Number of shoreline segments
MAXVERT	I6	Maximum number of shoreline segment vertices
SUBREC	A1	Subrecord delimiter (/)
(segments)		
NVERT	I6	Number of vertices in this segment
LNG	I6	Vertex longitude or X value
LAT	I6	Vertex latitude or Y value
(vertices repeat NVERT times)		
SUBREC	A1	Subrecord delimiter (/)
(segments repeat NSEG times)		
EOR	A1	